

# MOMOS-MT: Mobile Monophonic System for Music Transcription

## Sheet Music Generation on Mobile Devices

Munir Makhmutov  
AI in Games Dev. Lab  
Innopolis University, Russia  
m.makhmutov@innopolis.ru

Joseph Alexander Brown  
AI in Games Dev. Lab  
Innopolis University, Russia  
j.brown@innopolis.ru

Manuel Mazzara  
Service Science and  
Engineering Lab  
Innopolis University, Russia  
m.mazzara@innopolis.ru

Leonard Johard  
Service Science and  
Engineering Lab  
Innopolis University, Russia  
l.johard@innopolis.ru

### ABSTRACT

Music holds a significant cultural role in social identity and in the encouragement of socialization. Technology, by the destruction of physical and cultural distance, has led to many changes in musical themes and the complete loss of forms. Yet, it also allows for the preservation and distribution of music from societies without a history of written sheet music. This paper presents early work on a tool for musicians and ethnomusicologists to transcribe sheet music from monophonic voiced pieces for preservation and distribution. Using FFT, the system detects the pitch frequencies, also other methods detect note durations tempo, time signatures and generates sheet music. The final system is able to be used in mobile platforms allowing the user to take recordings and produce sheet music *in situ* to a performance.

### CCS Concepts

•Applied computing → Sound and music computing;  
•Information systems → Music retrieval; *Open source software*;

### Keywords

Monophonic Music Transcription; Music Analysis; Pitch Frequency Detection; Pitch Duration Detection; Tempo Detection; Time Signature Detection; Ethnomusicology

## 1. INTRODUCTION

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'17, April 3-7, 2017, Marrakesh, Morocco

Copyright 2017 ACM 978-1-4503-4486-9/17/04...\$15.00

<http://dx.doi.org/xx.xxxx/xxxxxxx.xxxxxxx>

The complexity and variety of musical genres is one of the major obstacles for the delivery of an effective commercial system for polyphonic music transcript. However, obstacles for its adoption are mostly of cultural nature and not technical. Music transcription is considered by musicians and music teachers a good pedagogical exercise, and there are prejudices in some communities about the liberalization of automatic tools. In some cases, though, written transcription is simply not possible, and this is mostly true for regional folk music that it is usually played and not written and the conservation process continued unchanged for centuries. In other cases, automatic music transcription would allow musicians to record the notes of an improvised performance that would otherwise get lost (this situation is typical of jazz) [6]. Software technologies can be of paramount importance in this context to simplify the diffusion of local music artifacts, protect them, and allow musicians in other parts of the globe to access them via Internet in a way that would have been unthinkable only two decades ago.

Automatic music transcription can be traced back to the late 1970s when Martin Piszczalski and Bernard Galler aimed at creating “a system that would automatically transcribe musical sounds into their equivalent music notation”. The objective of their project was to “automatically detect the notes in a played melody and write out the note patterns just as a human transcriber would” [29]. At the time, available technologies suggested the use of recorded tapes, and computers would analyze such tapes with music recorded from musical instruments. The researchers understood that their technique could have benefited several fields such as music education and music publishing. Their contribution, although among the first fully comprehensive from a methodological viewpoint, was not built in the vacuum. The idea of using a computer program to automatically convert sounds into written music appeared together with the widespread use of computers, although systematic methodologies took longer to appear [2]. Since the first studies on music transcription, one of the major challenges was the development of poly-

phonic systems, which appeared significantly more problematic. The seminal work by Moorer [27] was then continued by researchers at Stanford in the beginning of the 1980s [13, 12], still the systems did not allow the transcription of more than two polyphonic voices.

Despite the analogies with speech recognition, the commercial and academic interest for music transcription has been more moderate and has been mostly pushed by interdisciplinary research with broader goals than transcription itself. The automatic transcription of polyphonic music has been the subject of increasing interest during the last thirty years. Ideas that had originated in the 1970s encouraged research on the topics for the following decades and influenced different research fields, in particular coping with the limitation of monophonic transcription systems. Work has been done in multidisciplinary contexts too, for example in the crossover between robotics, AI and music transcription. One of the first projects allowing the transcription of more than two polyphonic voices started in the late 1980s at Osaka University in Japan and was aimed at extracting feelings from musical signals to form the software core of a robotic system that could respond to music as human listeners [26].

In this paper, we tackle the problem of automatic transcription of monophonic music via mobile device recording, and we present the tool MOMOS-MT (Mobile Monophonic System for Music Transcription).

Mobile devices are the major enabler for sheet music creation *in situ* to a performance. The authors believe this aspect represents the major driver of innovation and foresee a potential for commercialization to support the progress of ethnomusicology.

This paper is organized as follows. Section 2 describes in detail the problem description and the known techniques for pitch and tempo recognition. Section 3 is dedicated to the description of transcription system on which this paper is built. Section 4 reports the overall results.

## 2. PROBLEM DESCRIPTION

Methods and techniques for music transcription, in particular monophonic, have been investigated over four decades. In this section, we will present the literature in some detail for what concerns pitch (frequency and duration), tempo, and time signature detection.

### 2.1 Pitch Recognition Methods

There are a lot of methods used for pitch detection, but not all of them give good results [14, 19]. Some papers tell that autocorrelation can be used for monophonic transcription [9, 20, 23]. Some papers recommend to use Fourier transform for music transcription: for single notes and for chords. Statistics of these two methods usage show that autocorrelation has worse accuracy of pitch recognition than Fourier transform. Chord recognition is more difficult task, it also can be called polyphonic music transcription [3, 15].

The Fourier transform decomposes a function of time (*e.g.* a signal) into the frequencies that make it up. This process is similar to how a musical chord can be expressed as the amplitude or loudness of its constituent notes. The Fourier trans-

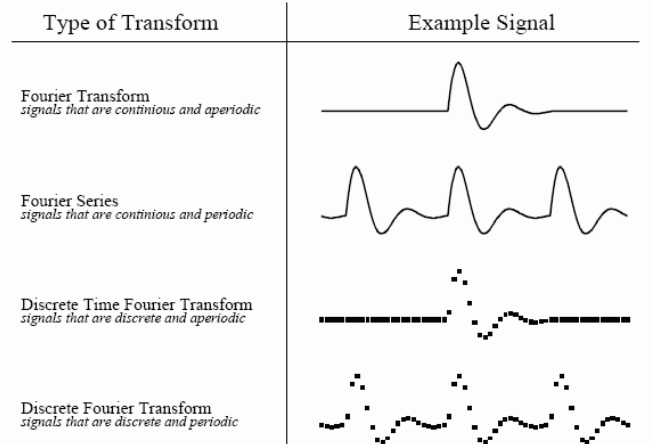


Figure 1: Fourier transform types [31]

form of a function of time itself is a complex-valued function of frequency, whose absolute value represents the amount of that frequency present in the original function, and whose complex argument is the phase offset of the basic sinusoid in that frequency. The term Fourier transform refers to both the frequency domain representation and the mathematical operation that associates the frequency domain representation to a function of time. The Fourier transform is not limited to functions of time, but in order to have a unified language, the domain of the original function is commonly referred to as the time domain. For many functions of practical interest, one can define an operation that reverses this: the inverse Fourier transformation, also called Fourier synthesis, of a frequency domain representation combines the contributions of all the different frequencies to recover the original function of time. There are four types of Fourier transform (Figure 1).

A fast Fourier transform (FFT) is an algorithm to compute the discrete Fourier transform (DFT) and its inverse. Fourier analysis converts time, or space to frequency, or wavenumber and vice versa; an FFT rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors. It is much faster than other Fourier transform types.

In his PhD thesis, Sterian [32] has used Pielemeier’s modal transform [28] as a front end, the peaks of which were tracked through frames by Kalman filtering for polyphonic transcription [25]. Reasonable success was found with up to four part brass ensembles (79.5% accuracy was given in four-note polyphony).

Walmsley used Markov Chain Monte Carlo (MCMC) methods to attempt the transcription task [33]. The number of notes and harmonics were unknown. This produced reasonable results, but with the inclusion of global parameters controlling the evolution of data over multiple frames, performance was improved. Godsill & Davy [21] continued this research and improved the model to take account of amplitude evolution within a single frame and also inharmonicity [8]. Their simulations are computationally intensive but

produce accurate results for traditionally difficult problems (*e.g.* fifths).

Recently attention has been directed to the transcription of non-Western music, which has underlined both the capability and limitations of current transcription systems [7] applied the polyphonic transcription techniques developed in [5] in order to transcribe traditional Turkish microtonal music. Cited accuracies are 56.7 % (F-measure). The same technique was applied by [1] to the the British Library World & Traditional Music Collections, but unfortunately accuracy estimate are not available. Another recent approach by [22] reached 57.9 % F-measure on Greek dances.

## 2.2 Pitch Frequency Precision

After recognition of note frequencies (no matter which recognition method was used), they should be changed. There are two reasons for changing of note frequencies:

1. musical instruments cannot be ideally tuned
2. even for well-tuned instrument pitch recognition methods cannot give ideal results

As previously mentioned, MIDI file can contain a sequence of notes that has one of 128 basic meanings of frequencies. As such, it is necessary to compare each frequency with these 128 basic frequencies and find the nearest. Of course, it is naive to compare each of the 128 pitches; it is better to use one of the optimization methods that will get the necessary pitch in a small amount of iterations.

The Dichotomy method can be used for this task because it can reduce the amount of iterations with a small amount of equations [34]. It is a method for numerically solving equations with a single unknown. Consider the equation

$$f(x) = 0$$

with a continuous function  $f$  on the interval  $[a, b]$  which takes values of different signs at the end points of the interval and which has a single root  $x^*$  within  $[a, b]$ . To find  $x^*$  approximately, one divides  $[a, b]$  into halves and calculates the value of  $f(x_1) = 0$  at the midpoint  $x_1 = (a + b)/2$ . If  $x_1 \neq 0$ , one takes the two intervals  $[a, x_1]$  and  $[x_1, b]$  and from them selects for the next dichotomy the one at the end points of which the values of the function differ in sign. This continued division into halves gives a sequence  $x_1, x_2, \dots, x_n$ , which converges to the root  $x^*$  with the rate of a geometrical progression:

$$|x_n - x^*| \leq \frac{b-a}{2^n}, n = 1, 2, \dots (1)$$

where the bound (1) cannot be improved upon in this class of functions. If  $f$  has more than one root in  $[a, b]$ , the sequence will converge to one of them.

A method for minimizing a function of one variable. One has to find the minimum

$$f^* = \min_{x \in [a, b]} f(x)$$

of a unimodal function  $f$  on an interval  $[a, b]$  and to determine the point  $x^*$  at which it is attained. For this, one

divides  $[a, b]$  into halves and near the middle  $\bar{x}_1 = (x + b)/2$  calculates the values of  $f$  at the two points  $x_1 = \bar{x}_1 - \varepsilon/2$  and  $x_2 = \bar{x}_1 + \varepsilon/2$ , where the number  $\varepsilon > 0$  is a parameter of the method and is sufficiently small. Then, the values  $f(x_1)$  and  $f(x_2)$  are compared, and on the basis that  $f$  is unimodal one selects from the two intervals  $[a, x_2]$  and  $[x_1, b]$  the one that certainly contains  $x^*$ . For example, if  $f(x_1) \geq f(x_2)$ , this will be  $[a, x_2]$ , otherwise  $[x_1, b]$ . The interval is again divided into halves, and near the middle  $\bar{x}_2$  one takes two points  $x_1 = \bar{x}_2 - \varepsilon/2$  and  $x_2 = \bar{x}_2 + \varepsilon/2$ , compares the values of the function, etc. As a result, one obtains a sequence of midpoints  $\{\bar{x}_n\}$ , for which

$$|x_n - x^*| \leq \frac{b-a-\varepsilon}{2^n} + \frac{\varepsilon}{2}, n = 1, 2, \dots, (2)$$

As an approximation to  $f^*$  the value  $f(\bar{x}_n)$  for sufficiently large  $n$  is taken.

The name is given to the method because at each step in this algorithm the segment containing the minimum becomes approximately half the length. The dichotomy method is not the best in the class of unimodal functions.

According to the number of MIDI levels (128), this algorithm will find the necessary frequency in seven-eight iterations. It is much better than 128 iterations. Other optimization methods also can be used. The disadvantage of other methods is that they are more difficult and will increase the amount of calculations, even the method of Golden section search, a type of modified Dichotomy method.

## 2.3 Tempo Detection

Tempo detection is one of the first tasks in music transcription. Mistake in tempo detection can lead to a high error probability of music transcription, which justifies our thoroughness in selecting methods for tempo recognition. One of the methods is Monte Carlo methods for tempo tracking [11]. Further, there are several open source libraries for different languages that can be used for this task. Tempo is not always constant in songs: it can vary throughout the song. The issue of tempo tracking is that it can be changed slowly and immediately. For this reason, tempo tracking can become a difficult task, especially when time signature changes as well. Most popular music is written in constant tempo; therefore, it is assumed in our method that songs have a constant tempo.

## 2.4 Time Signature Detection

As mentioned earlier, time signature can vary too, and it also can be constant during the song. Although non-constant time signature is not common for modern music, it is used in some songs. Detection of changing time signatures is not a trivial task. There are common signatures, such as quadruple (4/4) and triple (3/4) time. Most songs are written in this time and it is constant. However, some songs are written in less frequent time signatures, such as 5/4, 7/4, 9/4, etc. are more difficult to detect. Especially, when a time signature changes mid-song.

## 2.5 Pitch Duration Detection

Each note has its duration, and there are several approaches used for detecting pitch duration. One of these approaches

provides measuring of loudness and energy for each small part of song. So, higher loudness can mean that this part of song is represented by a new pitch. Of course, this method will work only if the previous note is considered. It is necessary because neighboring chunks of the song with the same meaning of pitch can be parts of one note or it can be two notes played consequently. If the second chunk has the same pitch and lower or same loudness, then it is most likely the same note. If the second chunk has the same pitch and higher loudness, then it is most likely two subsequent notes with the same pitch. Additional analysis is needed to make a decision about it.

### 3. TRANSCRIPTION SYSTEM

A transcription system prototype has been developed as a part of the research. It works mostly as a *proof of concept* in order to verify the potential for practical applications. Figure 2 describes the music transcription process as realized by the tool.

The music that has been recorded acts as input for the Music Analysis block. Here the recorded music is trimmed in order to detect tempo, time signature, bar amount, notes and their duration, which are all concepts explained in Section 2. Finally, and accordingly to the processed data, the tool generates a MIDI containing the information for creating sheet music. This sheet music represents the final output of the process. The prototype source code is available on GitHub at the URL: *hidden for peer review*. At the same URL, a set of recordings used to test the system is also available.

The key feature of the developed system is the ability to produce sheet music *in situ* to a performance. Using mobile devices has not only enabled, but also propelled the innovation for this prototype. The development has been driven by the definition of three principal use cases derived by the elicitation of functional requirements: music recording, music transcription, and transcription sharing. The use cases also describe the lifecycle of the mobile application and the the main steps of function.

#### 3.1 Requirements

This subsection enlists the necessary requirements that are required for the mobile monophonic music transcription system development. These requirements have been elicited from interactions with musicians, potential users, and specialists of Music Informatics from universities in Europe (*hidden for peer review*). Requirements from 1 to 5 have been fully implemented in the current prototype while the option of sharing created files via email is still under development and left as future work. At the same time, other sharing options are being considered under the wave of increasing use of social media by music experts.

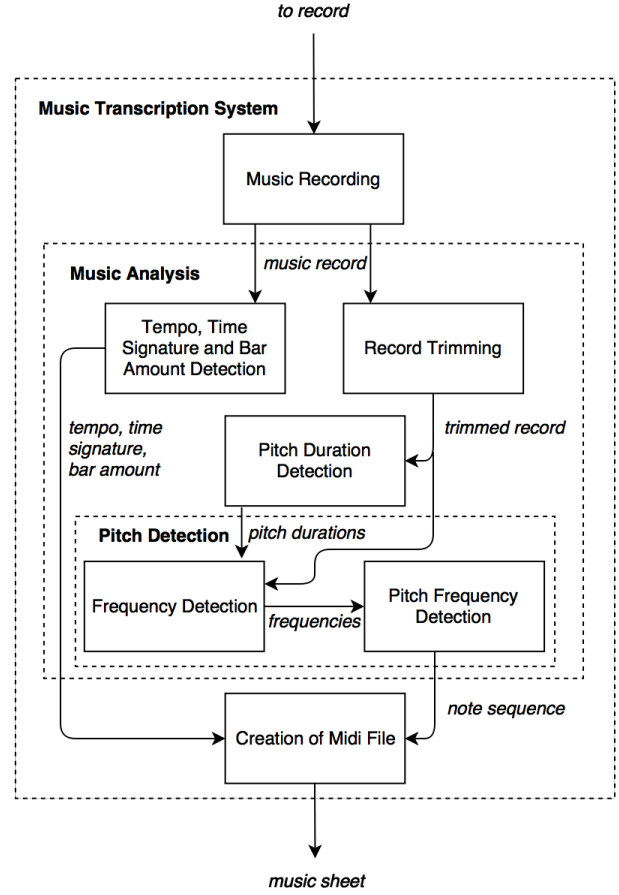


Figure 2: Music Transcription Process

| Requirement ID | FR-01   |
|----------------|---|
| Title          | Application should be available for mobile phones and tablets |
| Description    | Application should be available on Android devices            |
| Priority       | Mandatory (1)   |
| Risk           | Critical (C)  |

| Requirement ID | FR-02  |
|----------------|--|
| Title          | Application should allow user to record the music  |
| Description    | Application has to have an opportunity to record the music on given device via internal microphone |
| Priority       | Mandatory (1)  |
| Risk           | Critical (C)   |

|                       |   |
|-----------------------|---|
| <b>Requirement ID</b> | FR-03   |
| <b>Title</b>          | Application should allow user to transcribe monophonic music played by only one instrument  |
| <b>Description</b>    | Application should allow user to transcribe monophonic music with accuracy not less than 90%. Monophonic music provides single instrument that plays not more than one note per each moment |
| <b>Priority</b>       | Mandatory (1)   |
| <b>Risk</b>           | Critical (C)  |

|                       |  |
|-----------------------|--|
| <b>Requirement ID</b> | FR-04  |
| <b>Title</b>          | Application should allow user to save file with music transcripts. There are several file types that can be used: MusicXML, ASCII, MIDI, gpx |
| <b>Description</b>    | Application should save extracted music into file with popular extension that can be read and changed on computer                            |
| <b>Priority</b>       | Mandatory (1)  |
| <b>Risk</b>           | Critical (C)   |

|                       |  |
|-----------------------|--|
| <b>Requirement ID</b> | FR-05  |
| <b>Title</b>          | Application should eliminate excess noise from the record  |
| <b>Description</b>    | Application should eliminate excess noise in the record, which can lead to wrong music transcription |
| <b>Priority</b>       | Nice to have (2)   |
| <b>Risk</b>           | Medium (M)   |

|                       |  |
|-----------------------|--|
| <b>Requirement ID</b> | FR-06  |
| <b>Title</b>          | Application should allow user to send created file via email                     |
| <b>Description</b>    | Application should have menu option that can help to send created file via email |
| <b>Priority</b>       | Optional (3)   |
| <b>Risk</b>           | Low (L)  |

### 3.2 Use Cases

The use cases define the interaction between a human user (in this case) and the system, for achieving specific goals: *recording* music, *transcribe* a music file and *share* a music file via email. The three cases cover part of the functional requirements that have been elicited from musicians and experts for the system design. Use Case 3 is not implemented in the current prototype.

|                              |  |
|------------------------------|--|
| <b>Use case ID</b>           | UC-01  |
| <b>Title</b>                 | Record music file  |
| <b>Aim</b>                   | To make record with mobile/tablet  |
| <b>Reqs</b>                  | FR-1, FR-2, FR-5   |
| <b>Stakeholder interest</b>  | Music recording for further transcription  |
| <b>Precondition</b>          | 1. Application is launched<br>2. Enough free memory on device for record   |
| <b>Main success scenario</b> | 1. User chooses recording option<br>2. User records music via internal device microphone<br>3. Application eliminates excess noise from the record<br>4. User assigns name to the record<br>5. User saves record |
| <b>Post condition</b>        | Record is saved on device  |

|                              |  |
|------------------------------|--|
| <b>Use case ID</b>           | UC-02  |
| <b>Title</b>                 | Transcribe music file  |
| <b>Aim</b>                   | To transcribe record made with mobile/tablet   |
| <b>Reqs</b>                  | FR-3, FR-4   |
| <b>Stakeholder interest</b>  | Music transcription  |
| <b>Precondition</b>          | 1. Monophonic music file was recorded<br>2. Enough free memory on device for file transcription  |
| <b>Main success scenario</b> | 1. User chooses transcription option<br>2. User assigns name to the file with transcriptions<br>3. User saves file with transcriptions |
| <b>Post condition</b>        | Music transcription is done, file is saved on device   |

|                              |  |
|------------------------------|--|
| <b>Use case ID</b>           | UC-03  |
| <b>Title</b>                 | Send transcribed music file via email  |
| <b>Aim</b>                   | To send transcribed music file via email   |
| <b>Reqs</b>                  | FR-6   |
| <b>Stakeholder interest</b>  | Music transcription  |
| <b>Precondition</b>          | 1. Music transcription file was created<br>2. Internet connection exists   |
| <b>Main success scenario</b> | 1. User chooses option of sending file via email<br>2. User enters email address<br>3. User sends file via email |
| <b>Post condition</b>        | Music transcription is sent via email  |

### 3.3 Methods and Libraries

The transcription system was developed in Java by means of a number of libraries:

- *jMusic* (<http://explodingart.com/jmusic>) has been used for reading files in WAVEform audio format.
- *jEN* (<https://github.com/echonest/jEN>), an open source library, has been used for tempo, bar amount, time signature, and tempo tracking.

- *musicg* (<https://code.google.com/p/musicg>) has been used for record trimming (it was necessary to trim empty space at the beginning and at the end of records).
- FFT has been implemented by the open source library *Jtransforms* in combination with Dichotomy method have been used for notes' frequencies detection. Pitch duration detection is done by modified method of windowing<sup>1</sup>.
- The open source library *jFugue* (<http://www.jfugue.org>) has been used to generate MIDI-files based on results input analysis.

### 3.4 Running the System

We will show here how the system runs, *i.e.* how a piece of music recording leads finally to the generation of a music sheet. As examples, we consider two melodies that were used as inputs for MOMOS-MT.

The first melody (<https://goo.gl/a6juhn>) was played on a piano, while the second (<https://goo.gl/pP4n2i>) was played on a guitar. Both were recorded with a mobile phone recorder and outputs resulted in MIDI-files. The music was not transcribed taking into account the specific instruments (Output sample 1: <https://goo.gl/n0ZRZV>, Output sample 2: <https://goo.gl/ITeyV>). The transcription system does not indeed detect instruments, which is why both melodies were transcribed into piano batches. The instrument for the second output was manually changed to guitar in order to have an easier comparison. Post-transcription changes of the instruments do not alter pitches, time signatures, or tempo. The only difference is the sound of the instrument, or timbre.

Sheet music for the output and part of the second are shown in Figure 3 and Figure 4, respectively.

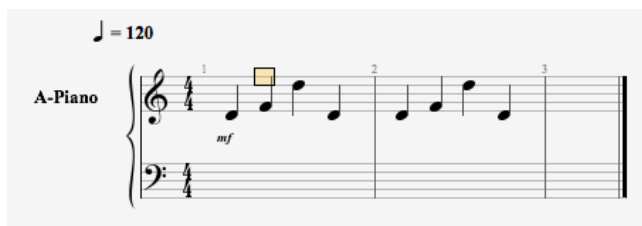


Figure 3: Output for the first record

## 4. CONCLUSIONS

In this paper, the problem of automatic transcription of monophonic music with the usage of mobile device recorder has been considered, and the tool MOMOS-MT has been presented. The use of mobile devices is the key enabler of sheet music creation *in situ* to a performance and represent the major driver of innovation and potential for commercialization of this prototype in the context of ethnomusicology.

Fast Fourier transform was proposed to detect pitch frequencies. We have discussed and studied the musical theory in order to examine the system's requirements. We have found

<sup>1</sup><https://sites.google.com/site/piotrwendykier/software/jtransforms>



Figure 4: Output for the second record

features, with which music can be described. The goal of the project was to create a mobile application for monophonic music transcription, which was achieved.

The system outputs may differ from the inputs. The cause of this discrepancy is currently under evaluation, and a systematic assessment is left as future work. Anecdotal evidence shows these errors produce correct notes, but with a one-octave shift without violating the melody, as may be the case with other potential error types.

This work has described a transcription method that can be easily extended in the future. The next step for this system is polyphonic music transcription. Also, this system can be extended in the field of musical instrument recognition [18, 10] and the detection of different playing techniques on different instruments. For example, tapping, vibrato [4], staccato, etc. Furthermore, dance music characteristics can be found, so the system will determine how popular transcribed music will be in disco clubs [17]. The direction of the sound also can be found, which can be useful for tracking systems [16, 24]. Marco Scirea *et al.* [30] created a tool for music generation. We plan future testing of MOMOS-MT to be performed taking as inputs the music generated by the tool. These future directions open new unexplored topics that will undoubtedly lead to new possibilities for all musicians of the world.

## 5. REFERENCES

- [1] S. Abdallah, A. Alencar-Brayner, E. Benetos, S. Cottrell, J. Dykes, N. Gold, A. Kachkaev, M. Mahey, D. Tidhar, A. Tovell, et al. Automatic transcription and pitch analysis of the british library world & traditional music collections. In *Proc of the 5th International Workshop on Folk Music Analysis-FMA2015, Paris, France*, pages 10–12, 2015.
- [2] A. Askenfelt. Automatic notation of played music (status report), 1976, speech transmission laboratory, quarterly progress attd status report, 1, royal institute of technology. 1976.
- [3] B. K. A.T. Cemgil and D. Barber. Generative model based polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to*

*Audio and Acoustics*, 2003.

- [4] S. M. Bendor D. Time domain extraction of vibrato from monophonic instruments. In *Proceedings of the 1st International Conference on Music Information Retrieval*, 2000.
- [5] E. Benetos and S. Dixon. A shift-invariant latent variable model for automatic music transcription. *Computer Music Journal*, 36(4):81–94, 2012.
- [6] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: challenges and future directions. *J. Intell. Inf. Syst.*, 41(3):407–434, 2013.
- [7] E. Benetos and A. Holzapfel. Automatic transcription of turkish microtonal music. *The Journal of the Acoustical Society of America*, 138(4):2118–2130, 2015.
- [8] G. Brown and M. Cooke. Perceptual grouping of musical sounds - a computational model. *J. of New Music Research*, 1994.
- [9] J. C. Brown and B. Zhang. Musical frequency tracking using the methods of conventional and "narrowed" autocorrelation. *The Journal of the Acoustical Society of America*, 89(5):2346–2354, 1991.
- [10] M. Buccoli, M. Zanoni, F. Setragno, F. Antonacci, and A. Sarti. An unsupervised approach to the semantic description of the sound quality of violins. In *23rd European Signal Processing Conference, EUSIPCO 2015, Nice, France, August 31 - September 4, 2015*, pages 2004–2008, 2015.
- [11] A. Cemgil and B. Kappen. Monte carlo methods for tempo tracking and rhythm quantization. *J. Artificial Intelligence Research*, 2003.
- [12] J. Chafe. Techniques for note identification in polyphonic music. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, 1986.
- [13] R. Chafe, Mont-Reynaud. Aesthetic response to choral music: Response comparisons of performer participants and non-performer respondents. *Computer Music Journal*, 6(1), 1982.
- [14] N. Cook. Perception: a perspective from music theory. *Musical Perceptions*, 1994.
- [15] J. B. Correa. Towards the automated analysis of simple polyphonic music: A knowledge based approach. 2003.
- [16] M. Davies. Audio source separation. In *Mathematics in Signal Processing V*, 2001.
- [17] S. Dixon. Classification of dance music by periodicity pattern. *ISMIR*, 2003.
- [18] A. Eronen and A. Klapuri. Musical instrument recognition using cepstral coefficients and temporal features. *ICASSP*, 2000.
- [19] W. Fucks. Mathematical analysis of formal structure of music. *IRE Trans. on Information Theory*, 1962.
- [20] M. S. Giuliano Monti. Monophonic transcription with autocorrelation. *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00)*, 2000.
- [21] S. Godsill and M. Davy. *Bayesian harmonic models for musical signal analysis*. PhD thesis, 7th Valencia Int. Meeting on Bayesian Statistics, 2002.
- [22] A. Holzapfel and E. Benetos. The sousta corpus: Beat-informed automatic transcription of traditional dance tunes. In *Proceedings of ISMIR - International Conference on Music Information Retrieval*, 2016.
- [23] M. S. Juan Pablo Bello, Giuliano Monti. Techniques for automatic music transcription. *ISMIR*, 2000.
- [24] M. K. *Sound-Source recognition*. PhD thesis, MIT, 1999.
- [25] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.
- [26] I. Katayose. The kansei music system. *Computer Music Journal*, 13(4):7277, 1999.
- [27] J. A. Moorer. *On the Segmentation and Analysis of Continuous Musical Sound by Digital COMPUTER*. Ph.D. thesis, Stanford University, 1975.
- [28] W. Pielemeier, G. Wakefield, and M. Simoni. Time-frequency analysis of musical signals. *Proceedings of the IEEE*, 84(9):1216–1230, 1996.
- [29] M. Piszczalski and B. A. Galler. Computer analysis and transcription of performed music: A project report. *Computers and the Humanities*, 13(3):195–206, 1979.
- [30] M. Scirea, J. Togelius, P. Eklund, and S. Risi. Metacompose: A compositional evolutionary music composer. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, pages 202–217. Springer, 2016.
- [31] S. W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, San Diego, CA, 1999.
- [32] A. Sterian. *Model-based Segmentation of Time-frequency Images for Musical Transcription*. PhD thesis, University of Michigan, Ann Arbor, 1999.
- [33] P. Walmsley. *Signal Separation of Musical Instruments - Simulation-based Methods for Musical Signal Decomposition and Transcription*. PhD thesis, Cambridge University Engineering Department, 2000.
- [34] D. Wilde. Optimum seeking methods. *Prentice-Hall*, 1964.